

LAB 8

MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Biết cách sử dụng Internal Storage.
- ✓ Biết cách sử dụng Shared Preferences.

NỘI DUNG

BÀI 1: Tiếp theo Lab7, sử dụng lưu, đọc dữ liệu (file text) trong Internal Storage để thực hiện chức năng đăng ký và đăng nhập. (sửa lại code của activity Register và Login)

Hướng dẫn:

- ❖ Tạo class **User** implements **Serializable**

```
public class User implements Serializable {  
    3 usages  
    private String username, password;  
  
    1 usage   1 Dinh Nguyen  
    public User(String username, String password) {  
        this.username = username;  
        this.password = password;  
    }  
  
    1 usage   new *  
    public String getUsername() {  
        return username;  
    }  
  
    new *  
    public void setUsername(String username) {  
        this.username = username;  
    }  
  
    1 usage   new *  
    public String getPassword() {  
        return password;  
    }  
  
    new *  
    public void setPassword(String password) {  
        this.password = password;  
    }  
}
```

❖ Lưu thông tin đăng ký vào file text:

```
public void writeUser(Context context, String fileName, User user) {
    List<User> list = new ArrayList<>();
    try {
        FileOutputStream fileOutputStream = context.openFileOutput(fileName, Context.MODE_PRIVATE);
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
        list.add(user);
        objectOutputStream.writeObject(list);
        objectOutputStream.close();
        fileOutputStream.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

❖ Đọc thông tin user từ file text:

```
public List<User> readUser(Context context, String fileName) {
    List<User> objectList = new ArrayList<>();
    try {
        FileInputStream fileInputStream = context.openFileInput(fileName);
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
        objectList = (List<User>) objectInputStream.readObject();
        objectInputStream.close();
        fileInputStream.close();
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
    return objectList;
}
```

BÀI 2: Sử dụng Shared Preferences thực hiện chức năng “Remember me” tại giao diện đăng nhập.

Hướng dẫn:

- ❖ Chỉnh sửa layout **activity_login**, thêm widget **CheckBox** và ánh xạ widget này.
- ❖ Tạo hàm để lưu dữ liệu vào **Shared Preferences** nếu **CheckBox** này được tích:

```
public void remember(String user, String pass, boolean chkRemember) {  
    SharedPreferences sharedPreferences = getSharedPreferences( name: "remember", MODE_PRIVATE);  
    SharedPreferences.Editor editor = sharedPreferences.edit();  
    editor.putString( s: "user", user);  
    editor.putString( s: "pass", pass);  
    editor.putBoolean( s: "chkRemember", chkRemember);  
    editor.apply();  
}
```

- ❖ Tạo hàm kiểm tra **CheckBox** có được tích hay không, khi khởi chạy activity:

```
public void checkRemember() {  
    SharedPreferences sharedPreferences = getSharedPreferences( name: "remember", MODE_PRIVATE);  
    String user = sharedPreferences.getString( s: "user", s1: "");  
    String pass = sharedPreferences.getString( s: "pass", s1: "");  
    boolean chkRemember1 = sharedPreferences.getBoolean( s: "chkRemember", b: false);  
    chkRemember.setChecked(chkRemember1);  
    if (chkRemember.isChecked()) {  
        edtUserName.setText(user);  
        edtPassword.setText(pass);  
    }  
}
```

BÀI 3: Lưu và đọc danh sách sinh viên trong activity AddStudent và activity Student từ file text, thực hiện sửa xoá danh sách sinh viên trong file.

Hướng dẫn:

- ❖ Đọc ra danh sách từ file:

```
public List<Student> getDataOld(String fileName){
    List<Student> list = new ArrayList<>();
    try {
        FileInputStream fileInputStream = this.openFileInput(fileName);
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
        list = (List<Student>) objectInputStream.readObject();
        objectInputStream.close();
        fileInputStream.close();
    } catch (Exception e){
        e.printStackTrace();
    }
    return list;
}
```

- ❖ Thêm dữ liệu vào list cũ và lưu lại vào file: với cách làm này chúng ta có thể lưu 1 list nhiều sinh viên, còn thực hiện như bài **Login**, thì khi tạo user mới user đã tạo trước đó sẽ bị ghi đè lên.

```
public void writeStudent(Context context, String fileName, Student student){
    List<Student> list = getDataOld(fileName);
    try {
        FileOutputStream fileOutputStream = context.openFileOutput(fileName, Context.MODE_PRIVATE);
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
        list.add(student);
        objectOutputStream.writeObject(list);
        objectOutputStream.close();
        fileOutputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- ❖ Tạo 2 interface để thực hiện việc sửa xóa list trong file:

```

1 package com.example.lab4_ps23456;
2
3 public interface UpdateItem {
4     void onClickButtonUpdate(int position);
5 }
6

```

- ❖ Đưa 2 interface này vào **AdapterListView**, điều này đồng nghĩa chúng ta sẽ không tương tác với **listview** trực tiếp trong **adapter** mà sẽ tương tác thông qua **interface**:

```

2 usages
DeleteItem deleteItem;
2 usages
UpdateItem updateItem;

1 usage
public AdapterListView(List<Student> list, Context context, DeleteItem deleteItem, UpdateItem updateItem) {
    this.list = list;
    this.listOld = list;
    this.context = context;
    this.deleteItem = deleteItem;
    this.updateItem = updateItem;
}

```

- ❖ Trong hàm **getView()** của **adapter** truyền **position** vào cho hàm của **interface**:

```
btn_update.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        updateItem.onClickButtonUpdate(i);  
    }  
});  
  
btn_delete.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        deleteItem.onClickButtonDelete(i);  
    }  
});
```

- ❖ Chuyển qua activity **Home** thực hiện chức năng **xoá**, đầu tiên cần xoá phần tử trong **list**, sau đó ta lại lưu **list** đó vào lại **file**: (lưu ý tên file)

```
public void delete(List<Student> list,int i){  
    list.remove(i);  
    try {  
        FileOutputStream fileOutputStream = Home.this.openFileOutput("student.txt", Context.MODE_PRIVATE);  
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);  
        objectOutputStream.writeObject(list);  
        objectOutputStream.close();  
        fileOutputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

- ❖ Truyền tham số vào cho **adapter**, xử lý sự kiện xoá, sự kiện update sinh viên tự thực hiện:

```
adapterListView = new AdapterListView(list, context: this, new DeleteItem() {
    1 usage
    @Override
    public void onClickButtonDelete(int position) {
        AlertDialog.Builder builder = new AlertDialog.Builder(context: Home.this);
        builder.setTitle("Xoá sinh viên");
        builder.setMessage("Bạn có chắc chắn muốn xoá sinh viên này không?");
        builder.setPositiveButton(text: "Xoá", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                delete(list, position);
                adapterListView.notifyDataSetChanged();
            }
        });
        builder.setNegativeButton(text: "Huỷ", listener: null);
        builder.show();
    }
}, new UpdateItem() {
    1 usage
    @Override
    public void onClickButtonUpdate(int position) {
        // hãy thực hiện update sinh viên tại đây
    }
});
lv_list.setAdapter(adapterListView);
adapterListView.notifyDataSetChanged();
```

- **Sau khi hoàn tất Lab8 hãy view lại Lab4 đến Lab8 để xem lại thành quả của các bạn, các bạn đã tạo ra được app mini, chúc các bạn học tốt!!!**

BÀI 4: GV CHO THÊM

*** YÊU CẦU NỘP BÀI:

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết ---